PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 3/12 — A1

(11) International Publication Number: WO 96/39656

(43) International Publication Date: 12 December 1996 (12.12.96)

(21) International Application Number: PCT/US96/08491

(22) International Filing Date: 3 June 1996 (03.06.96)

(30) Priority Data:

08/470,961	6 June 1995 (06.06.95)	US
08/470,962	6 June 1995 (06.06.95)	US
08/470,969	6 June 1995 (06.06.95)	US

(71) Applicant: APPLE COMPUTER, INC. [US/US]; 1 Infinite Loop - MS: 38-PAT, Cupertino, CA 95014 (US).

(72) Inventors: WALKER, Ted, W.; 857 West Remington Drive, Sunnyvale, CA 94087 (US). REILLY, Paul, E.; 6608 Creekview Court, San Jose, CA 95120 (US).

(74) Agents: CARMICHAEL, Paul, D. et al.; Apple Computer, Inc., 1 Infinite Loop - MS: 38-PAT, Cupertino, CA 95014 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

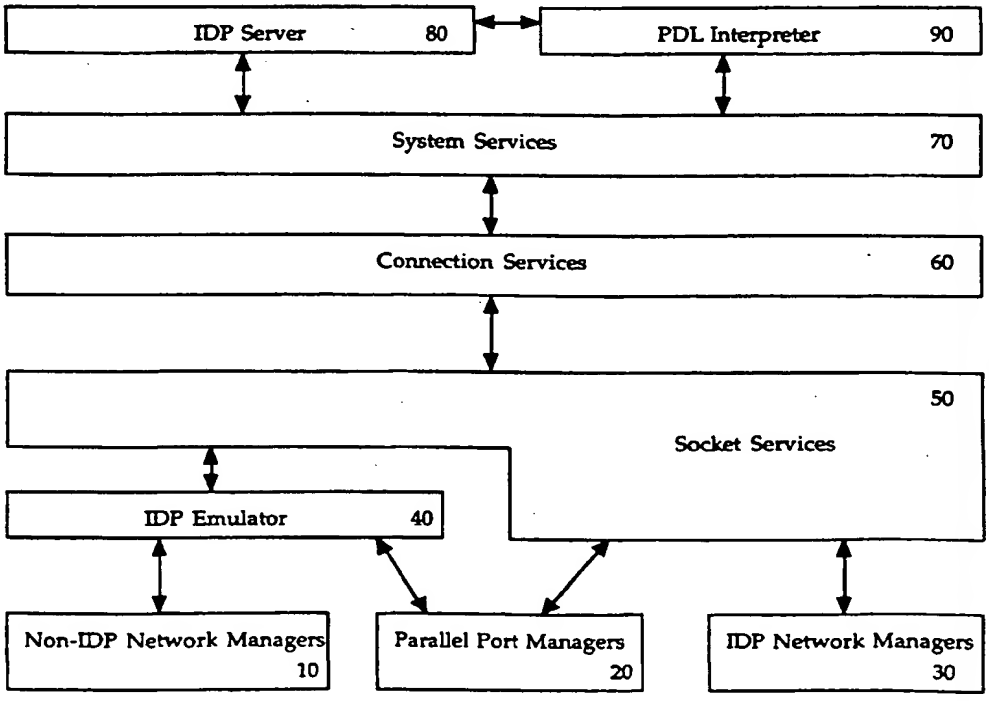
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: ARCHITECTURE FOR NETWORK PRINTING SYSTEMS

(57) Abstract

A printing system is provided for enhancing two-way communication between host computers and printers connected to a network. The network printing system includes an imaging device protocol (IDP) and an IDP emulator which enable various network service protocol/ports for host computers to communicate with the printer. The printing system also includes a client server connection protocol (CSCP) that facilitates connectivity by providing a symmetric protocol for connections between servers and clients over connection-based stream-oriented protocols such as AppleTalk Data Stream Protocol (ADSP) and Transmission Control Protocol (TCP). The dynamic port negotiation of CSCP facilitates concurrency for server applications by off-loading the port negotiation that would otherwise have to take place by the server and its clients. Thus, IDP and the IDP emulator enable various network service protocol/ports, both IDP and non-IDP ports, to communicate with the network printer by "seamless plug and play" connectivity. Host computers request network printer services through CSCP and IDP when these protocols are supported and emulates CSCP/IDP interactions when the network services do not support the CSCP/IDP protocols so that print job requests may be executed from the same print queue.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	**GB**	United Kingdom	**MW**	Malawi
AT	Austria	**GE**	Georgia	**MX**	Mexico
AU	Australia	**GN**	Guinea	**NE**	Niger
BB	Barbados	**GR**	Greece	**NL**	Netherlands
BE	Belgium	**HU**	Hungary	**NO**	Norway
BF	Burkina Faso	**IE**	Ireland	**NZ**	New Zealand
BG	Bulgaria	**IT**	Italy	**PL**	Poland
BJ	Benin	**JP**	Japan	**PT**	Portugal
BR	Brazil	**KE**	Kenya	**RO**	Romania
BY	Belarus	**KG**	Kyrgystan	**RU**	Russian Federation
CA	Canada	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CF	Central African Republic			**SE**	Sweden
CG	Congo	**KR**	Republic of Korea	**SG**	Singapore
CH	Switzerland	**KZ**	Kazakhstan	**SI**	Slovenia
CI	Côte d'Ivoire	**LI**	Liechtenstein	**SK**	Slovakia
CM	Cameroon	**LK**	Sri Lanka	**SN**	Senegal
CN	China	**LR**	Liberia	**SZ**	Swaziland
CS	Czechoslovakia	**LT**	Lithuania	**TD**	Chad
CZ	Czech Republic	**LU**	Luxembourg	**TG**	Togo
DE	Germany	**LV**	Larvia	**TJ**	Tajikistan
DK	Denmark	**MC**	Monaco	**TT**	Trinidad and Tobago
EE	Estonia	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**UG**	Uganda
FI	Finland	**ML**	Mali	**US**	United States of America
FR	France	**MN**	Mongolia	**UZ**	Uzbekistan
GA	Gabon	**MR**	Mauritania	**VN**	Viet Nam

ARCHITECTURE FOR NETWORK PRINTING SYSTEMS

TECHNICAL FIELD

The present invention is related to an architecture for a network printing system.

BACKGROUND

As the size of networked systems increases from a few connected computers located near each other to several thousand machines of various types and sizes scattered at great distances from each other, the demands on the network printers also increase. One of the most challenging demands in the implementation of network printers is connectivity. Typically, the network printer is a stand-alone peripheral device which is desired to perform in an increasing number of network and non-network communication environments. In network environments, the printer is desired to provide "seamless plug and play" connectivity for the continuous expansion of network service protocol/ports. For example, known network service protocol/ports that are desired to communicate with network printers are: printer access protocol (PAP)/LocalTalk (LT); PAP/EtherTalk (ET); transmission control protocol/internet protocol (TCP/IP); TCP/IP port 9100; LPR/TCP/IP; Novell remote printer or print server modes; bidirectional or unidirectional Centronics; AppleTalk Data Stream Protocol (ADSP)/EtherTalk; ADSP/LocalTalk; and Novell sequence packet exchange (SPX). These are just a few examples of the constantly increasing number of network protocol/ports that desire "plug and play" connectivity to the network printer.

A printer is a peripheral device that can be shared on a network connected to a print server. Also, a printer can directly connect to the network by including a network interface card or the network interface into the printer which allows the printer to run its own print server software and function as a regular node. In a typical Windows environment, the network printer is a one-way printer. In a one-way printer, the only communication path is from the host computer to the printer. As a result, the data is sent from the host computer to the printer without any ability for the printer to notify the host computer of the job status such as whether the print job has been successfully completed.

The use of a two-way printer has been developed for Macintosh Computer systems (Apple Computer, Cupertino, California). In the AppleTalk protocol used by the Macintosh systems, PAP is the protocol used for

communication between the Macintosh host computers and printers. PAP sets up, maintains and terminates the connection between the host computer nodes and the network printer and also transfers the data from the host computers to the printers. PAP allows two-way communication from the network printer to the Macintosh host computer to the extent that the status of the network printer is sent to the Macintosh host computer through PostScript. However, most page description language (PDL) interperters only allows one job to be processed at one time which prevents asynchronous interaction between multiple clients and the network printer.

Typically, networks utilize shared resource printers. Figure 1 illustrates the use of a conventional shared resource printer in a network. A plurality of host computers 1_0, 1_1,...1_n and a printer 5 are connected in a network. The printer 5 includes a disk 6 connected internally or externally to the printer 5 for storing the print jobs sent from the host computers $1_{0..n}$ before being printed by the printer 5. In shared resource printers, the host computer that first sends a print job to the network printer will have its job printed. Other host computers attempting to have a print job printed by the network printer while another job is being printed will not be able to access the network printer. Therefore, these host computers will repeatedly try to access the network printer during a timing scheme until the network printer becomes available. As a result, the speed of the network will be reduced by the network traffic due to the continuous polling of the host computers to the network printer during the time that the network printer is unavailable. This polling process is undesirable since a host computer will gain access to the printer, with only limited and possibly inexact arbitration after the print job is completed if a number of host computers are attempting to gain access while the printer is busy.

Traditionally, print servers provided access to the network printers and ran the programs necessary to create and operate a print queue for jobs sent to the printers from the host computers. The print queue is a directory that stores into the disk 6 the print jobs waiting to be printed. The print jobs are then printed from the print queue in a FIFO sequence. The print servers require storing the job request and entire job data by spooling this data to the disk 6. Depending on the job size, millions of bytes of data may be spooled to the disk and therefore a large sized disk is necessary.

Even if a large sized disk is used, the disk capacity for storing print jobs may be exceeded when a number of host computers send print jobs to the printer at approximately the same time. When the capacity of the disk is

reached, a back up procedure will be implemented where the printer will be polled every two seconds until space in the disk becomes available in a similar manner as in the shared resource printer.

Print servers and a printer may be combined in the same machine on many networks for economical reasons. The main advantage for combining the printer and print servers is that files do not need to be sent from the host computer to the print server machine and then from there to the printer in such a configuration. However, the disadvantage of combining the printer and print servers is that the required control for the print queue and the printing activity takes away CPU time from other network activity. In either case, a disk is required to store the data for the entire print job and since the print job is sent from the host computer, spooled to a disk and then transmitted to the network printer when the printer becomes available, unnecessary network activity results.

SUMMARY

To address the foregoing limitations associated with prior art systems, the present invention provides printing systems in accordance with independent claims 1 and 19, methods in accordance with independent claims 5, 15, 23, an architecture in accordance with independent claim 9, a printer in accordance with independent claim 14, and a computer readable medium in accordance with independent claim 18. Further advantageous features, aspects and details of the invention are set forth in the dependent claims, the following description and the drawings. The claims are to be understood as a first non-limiting approach of defining the invention in general terms.

These and other objectives are achieved in accordance with the present invention by a component based architecture which provides a comprehensive structure to a plurality of protocol/ports that are desired to operate on the network. In an exemplary embodiment of the invention, the architecture provides an imaging device protocol (IDP) and an IDP emulator which enable a wide variety of network services to uniformly interface with the printer. The layered architecture allows the uniform interface to the plurality of network services, which are both IDP and non-IDP services, and also permits new protocol/ports to be easily extended and integrated.

As a further feature of the invention, the layered architecture allows the network printer to achieve a "seamless plug and play" connectivity for the variety of protocol/ports. The layers of the architecture are divided so that each layer provides a normalized interface to the layers above until the top

layer of abstraction is reached in which all protocol/ports of network services interact in the same way.

Furthermore, in an exemplary embodiment of the invention, a client server connection protocol (CSCP) facilitates the client/server connections by allowing CSCP clients to request passive or active connections for specific services on a specific node and then the negotiation of port/sockets takes place over a CSCP control stream. Thereby, a dynamic port negotiation of the CSCP facilitates concurrency for server applications with a hierarchal structure of routines (a CSCP stack) by off-loading the port negotiation that would otherwise have to take place by the server and its clients.

As a further feature of the invention, CSCP also provides the capability for a server to reconnect to a previously connected client. As a result, a printer may call back and initiate a connection with a host computer through CSCP. CSCP allows printers and the host computers to interact in a symmetrical client/server model wherein either side may act as a server or a client in contrast to conventional printer implementations where only the host could initiate connections to the printer in such a symmetrical model.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will become more fully understood from the detailed description given hereinbelow and the accompanying drawings which are given by way of illustration only, and thus are not limitative of the present invention, wherein:

Figure 1 illustrates a network using a conventional printing system;

Figure 2 illustrates the layers of a network architecture for an embodiment of the present invention;

Figure 3 illustrates a more detailed representation of the network architecture for an embodiment of the present invention;

Figure 4 illustrates a process for enabling connections between clients and a printer in an embodiment of the present invention;

Figure 5 illustrates a process for enabling connections between non-IDP clients and a printer for an embodiment of the present invention;

Figure 6 illustrates the network configuration of a printing system for an embodiment of the present invention;

Figure 7 illustrates a control connection of a host initiated client/server sequence used for a client/server dialogue in an embodiment of the present invention; and

Figure 8 illustrates a new control connection of a host initiated client/server sequence used for a client/server dialogue in an embodiment of the present invention.

DETAILED DESCRIPTION

Figure 2 provides an illustration of the layers of a network architecture in an embodiment of the present invention. Each of these layers provides a normalized interface with the lower layers and their relationship will be described from the lower layers, which interact with the clients, to the upper layers where all of the protocols/ports interact with the printer in essentially the same way. The interfacing of these layers may be implemented in one or more static memory devices, such as a ROM, which is installed within the network printer.

At the lower layer of the architecture as illustrated in Figure 2, are non-imaging device protocol (IDP) network managers 10. The non-IDP network managers 10 control network ports that communicate with protocols other than IDP. Examples of some non-IDP protocols are: PAP/LocalTalk (LT), PAP/EtherTalk (ET), TCP/IP port 9100, LPR/TCP/IP, Novell remote printer or print server modes and bidirectional or unidirectional Centronics. The non-IDP network managers 10 support these protocols to be backward compatible with existing communication protocols. The non-IDP clients cannot access all of the enhanced IDP features in the upper layers, which will be described in more detail, but these non-IDP clients will be able to enter the print queue for processing and will be fairly arbitrated with other print jobs from IDP clients.

IDP network managers 30 provide an interface for IDP protocol/ports. Some examples of protocol/ports that are presently configured for IDP mode communication include ADSP/EtherTalk, ADSP/LocalTalk, SPX (Novell), TCP/IP and bi-directional Centronics. These protocol/ports are able to communicate with all of the IDP features which enhance the two-way communication between the host computers and the network printer. However, many other existing protocol/ports may be modified to support IDP and new protocols/ports may be created to support IDP. The interface of the IDP network managers with these IDP protocol/ports is preferably accomplished with a modified version of the Berkeley socket interface.

The Berkeley socket interface is a socket standard implemented in Berkeley Software Distribution (BSD) UNIX 4.3. (Interworking With TCP/IP, Vol. 3, Comer et al., Prentice Hall, 1993, pp. 49-55). The conventional Berkeley socket interface has the following major calls: a socket call for creating a socket;

a bind call for binding a socket to an end point of a known protocol/port; a listen call for placing a socket bound to a known port in a passive mode; an accept call for accepting incoming connections from a passive socket; a connect call for opening active connections to remote endpoints; a read call for reading data from existing connections; a write call for writing data to existing connections; and a close call for closing existing connections.

In an embodiment of the present invention, the Berkeley socket interface has simplified the major calls to a passive open call for combining the functionality of the conventional socket, bind, and listen commands, and an active open call for combining the functionality of the conventional socket and connect calls. The conventional accept, read, write and close calls have simply been renamed for the modified version of the Berkeley socket interface in the present embodiment.

A parallel port manager 20 provides both a non-IDP UNIX I/O interface and an IDP capable socket interface. The parallel port manager 20 is designed to support a host computer port which alternates between using IDP and non-IDP drivers/utilities to interact with the network printer. An example of such a port is a bi-directional Centronics port. IDP mode communication over Centronics ports should take place using a session/transport/network layer protocol. One example of such a session/transport/network layer protocol is a socket simulation protocol (SSP) for allowing the Centronics port to be used in the same way as other network ports. The parallel port manager 20 is able to automatically sense the transitions between SSP and non-SSP data over Centronics ports.

An IDP emulator 40 enables connections over non-IDP ports so that the upper layers may uniformly interface with both IDP and non-IDP clients. The IDP emulator 40 monitors connection requests on all non-IDP protocol/ports from the non-IDP network managers 10 and the parallel port manager 20 and emulates IDP requests as the connection requests are received. When a connection request is received from a non-IDP client, the IDP emulator 40 builds an IDP print job request message. The IDP emulator 40 ensures that print jobs from non-IDP protocol/ports are arbitrated fairly. The IDP emulator 40 allows non-IDP print jobs to be queued with IDP print jobs so that all of the print jobs may be processed in the order from which the connection attempts were received.

A socket services layer 50 is the interface component to all socket interface I/O managers. Sockets are general purpose interprocess communication mechanisms which are logical entities through which

programs or processes communicate with the network. Sockets are typically supported in libraries by UNIX implementations for operating systems such as DOS or OS/2 and for network operating systems such as Novell's netware and AppleTalk. A socket interface is a set of functions associated with a particular network node and when a client interfaces with the socket, information and network services may be requested and received.

The connection services layer 60 includes a client server connection protocol (CSCP) library which includes a hierarchal structure of routines (a CSCP stack). CSCP is a symmetric protocol that facilitates connection between servers and clients over connection-based stream-oriented protocols such as ADSP, TCP and SPX. CSCP resides in the presentation layer of the OSI seven-layer model for interfacing with one or more session level protocols. CSCP facilitates client and server connections by allowing clients of CSCP stacks to request passive or active connections for specific services on a specific node and specifies the service using an alpha-numeric string. Server applications, such as IDP, will minimally open a passive connection through the CSCP stack which specifies the service provided with a service specifier string. Clients of the server open an active connection through CSCP by specifying an identical service specifier string. CSCP implementations on the client and server nodes then negotiate the ports or sockets over which the client/server connection will take place, establish the connection, and return to their clients. Once the client/server connection is achieved, CSCP functions are completed.

Furthermore, CSCP facilitates reconnection to previously connected hosts. IDP uses the reconnection mechanism to implement remote queuing features, to subsequently request job data, and to send asynchronous status updates to clients which support IDP. More particularly, IDP only stores the job data request at the printer when the printer is busy printing another job, and the actual job data will remain locally at the host computer. Thereafter, the network printer will call back the host computer which corresponds to the first job data request queued in the print queue after the current print job is completed.

The negotiation of port/sockets takes place over a CSCP control stream. The CSCP implementation on the client side opens a control connection to the CSCP control port on the server's node. The control port may either be a known port of a port previously specified in a CSCP transaction. Once the control connection is established, port negotiation takes place, and the control connection is either closed or used as the client/server connection. A CSCP implementation may or may not accept more than one concurrent control

connection, depending on the nature of the services it enables, and expected usage of those services. For example, a laser printer may concurrently interact with several host computers, which requires concurrent processing of control connections, while a host computer may expect to interact with only a single printer at a time, which allows the host computer to process control connection requests iteratively. Figures 7 and 8 illustrate examples of host initiated client/server sequences for embodiments of the present invention. In Figure 7, a control connection used for a client/server dialogue is illustrated. In Figure 8, a new connection for a client/server dialogue is illustrated.

In an example of a packet format for CSCP packets, a header of data commonly includes protocol or revision data, packet length data, request sequence number data, message data, and message class data. The protocol revision data enables CSCP implementations to recognize and handle previous protocol revisions from other nodes when the protocol is revised. The packet length data defines the length of the entire CSCP packet. The request sequence number data is defined and incremented by a CSCP implementation when a request is made and is echoed in response to the request. Thereby, CSCP implementations are able to identify responses to multiple pending requests made to other nodes. The message data contains one of a plurality of message codes and a message class data contains CSCP request, CSCP response, or CSCP unsolicited commands. The CSCP request indicates that a request has been made, the CSCP response command indicates that the packet is a response to a request, and the CSCP unsolicited command indicates if the packet is providing unsolicited information. A CSCP request service port message is an example of message data that may be included in the CSCP packet. The CSCP request service port message is in the CSCP request class and is used to initiate port negotiation when the CSCP implementation receives an active connected request from a client. An option code is included within this message that specifies the action to take if the specified server is not immediately available. If the specified service is not immediately available, the server CSCP implementation will return a CSCP wait for callback command and close the control connection but will reconnect to the client CSCP control port at a later time when the service becomes available. At this time, the server CSCP implementation will send a CSCP callback message to the client CSCP implementation.

CSCP request service port message is another message that may be included within the message data of the CSCP packet. The CSCP request service port message is in the CSCP response class and is a response to the

CSCP request service port message. The specific data contains the response code. If the response code is CSCP service found, the message specific data will also contain the port/socket number, to which the requesting CSCP implementation should connect for the specified service. If the response code contains CSCP current connection, the current control connection will be used for the client/server connection. A response of CSCP unknown service will be returned if no matching service is found and a response of CSCP wait for callback will be returned when the CSCP callback when available option is used in the request and no matching service is immediately found. In response to the CSCP wait for callback response, the CSCP client should close the connection and wait for a callback message from the server.

Another example of a message that may be included within the message data of the CSCP packet is a CSCP callback message within the CSCP request class. The CSCP callback message is sent by a CSCP server to a CSCP client to complete disposition of a previous service request that was pending with a CSCP wait for callback response. The disposition code will either be CSCP service found or CSCP unknown service, if the CSCP server exceeds a predetermined time limit while waiting for a matching server application. If the disposition code is CSCP service found, the CSCP client should match the sequence number in the header to identify the original request and respond to the CSCP server.

A further example of a message contained within the message data of the CSCP packet is a CSCP callback message which is in the CSCP response class. If the disposition code in the request message is CSCP unknown service, the response code should be CSCP acknowledged. If the disposition code is CSCP service found, the response code should be either CSCP use current connection when the client/server dialogue is to take place on the concurrent connection, or CSCP use new port, when the CSCP client wants the client/server dialogue to take place on a new connection. In the case for a new connection, the CSCP client executes a passive open and then passes the new port/socket number from that passive open to the CSCP server in the port/socket number parameter. The server subsequently connects to that port/socket for the client/server dialogue.

The system services layer 70 provides a uniform interface to all of the IDP and non-IDP network services. In a preferred embodiment, the system services layer 70 is an interface which is an extension of the UNIX I/O interface. Examples of the functions performed by the system services layer 70 are: open; close; reconnect; read; write; swrite; and ioctl. The swrite function is

unique for the present embodiment. The swrite function enables a status description string to be passed to the I/O managers and the I/O managers may use or dispose of the status depending on the requirements of the protocol/port being managed.

The system services layer 70 also provides a buffered I/O layer for grouping read and write data into blocks of data which may be processed more efficiently. For example, buffers may be provided for 4K bytes of data. Thereby, the number of times that data is handed back and forth is reduced by eliminating the repeated processing of individual data and grouping the data in a larger block that may be easily processed.

The system services layer 70 interfaces with an IDP server 80 and a page description language (PDL) interpreter 90. The IDP server 80 is an application layer network protocol for enhancing the two-way communication between a host computer and the network printer. The IDP server 80 is independent of the network layers below and only requires that the transport be bidirectional. The IDP server 80 allows access to shared information in the network printer and enables two-way communication between the network clients and the network printer to exist at an enhanced level.

Some of the enhanced two-way communication provided by the IDP server 80 includes reading and modifying system parameters, adding and removing fonts, submitting print jobs and manipulating the print queue. By enabling clients of the printer to make job requests which are entered in the print queue without sending the actual print job data, an enhanced set of services for submitting print jobs, retrieving print data queue and asynchronously notifying clients with job status may be achieved. The PDL interpreter 90 may be provided for supporting PostScript, for example. Additional PDL interpreters may be provided to support other languages as desired. The presentation level data from non-IDP protocols/ports is assumed to be PDL data in the example configuration for the present embodiment. The non-IDP network managers 10 and the parallel port managers 20 may be implemented with a UNIX standard I/O interface in a preferred embodiment. The IDP emulator 40 then handles all of the non-IDP protocols/ports, provides a socket interface and emulates the CSCP/IDP interaction necessary to execute print job requests.

In one example of a header format for IDP messages, the structure includes protocol, command, result code, message byte length, and data fields. IDP is a set of related protocols and the protocol suites are intended to map different cooperating servers that add value to a printing environment. A

message can be initiated by either a host computer or the network printer. Currently, IDP includes a core protocol, a print protocol, and a font protocol. The core protocol contains commands that are common to all servers and are independent of their type. Printers may support the print protocol and font servers may support the font protocol. Common printers maintain fonts locally and therefore support part of the font protocol as well. As new type of servers are identified, additional protocols can be added to the suite of related protocols for IDP. The protocol field indicates which protocol of the IDP suite that the message is intended for. The command field indicates which command that the message corresponds to. The result code is a result of the command that is ignored on request commands. The message byte length field is the number of data bytes in the message. The data field includes message specific data and parameters.

Figure 3 illustrates a more detailed example of the network services architecture for an embodiment of the present invention. The non-IDP network managers 10 correspond to a PAP/ET manager 11, a PAP/LT manager 12, a Novell stream manager 13, a LPR manager 14, and a TCP port 9100 manager 15 in this example. The parallel port manager 20 includes a Centronics manager 21, and the IDP network managers 30 include a TCP/IP manager 31, an ADSP manager 32, and a Novell socket manager 33 in this example. The parallel port manager 20 also includes a Centronics autosense manager 22 for detecting between non-SSP and SSP data sent over the Centronics port, and a socket emulator 23 handles SSP communication. The IDP emulator 40 directs the non-IDP data to the appropriate non-IDP network manager 10. The connection services layer 60 is shown to include a connection manager 61 and a CSCP library 62 for providing the symmetric protocol that facilitates connection between servers and clients over connection-based stream-oriented protocols.

The system services layer 70 is connected to the connection manager 61. The system services layer 70 provides interaction between the IDP server 80 and the PDL interpreter 90. The IDP server 80 is shown in Figure 3 to include a print queue 82 and a job dispatcher 84. The PDL interpreter 90 interacts with the IDP server 80 when the job dispatcher 84 dispatches a job from the print queue 82. The information from the print queue 82 proceeds through the layers to connect with a client and obtain the actual print job data that will proceed from the client back up through the layers to the PDL interpreter 90 for printing at the network printer.

Figure 4 illustrates a process for enabling connections between IDP clients and the printer for an embodiment of the present invention. At step 600, an IDP host and IDP network manager opens a connection to a well known CSCP port. The connection is received by the connection services layer 60 via the socket services layer 50 at step 602 and the IDP host sends a CSCP request service port request at step 604. The connection services layer 60 receives the CSCP request at step 606 and a pending open from the IDP server 80 is received from the connection services layer 60 via the system services layer 70 at step 608. The IDP host issues a print job request to the IDP server 80 at step 610 and the IDP server 80 queues the print job in the print queue 82 at step 612 before closing the connection at step 614. The system services layer 70 dequeues the print job request at step 616, and issues a reconnection request at step 618. The connection services layer 60 issues an active open to the host via the socket services layer 50 and the I/O manager 30 at step 620. The connection services layer 60 also sends a CSCP request service port to the host at step 622. At step 624, the connection services layer 60 returns from a reconnection request via the system services layer 70. The system services layer 70 issues "a send job data" IDP request to the host at step 626 and then the systems services layer 70 sends the I/O stream to the PDL interpreter 90 at step 628. The host sends and receives data to and from the PDL interpreter 90 at step 630 and the host closes the connection at 632 when the sending and receiving of data is completed.

Figure 5 illustrates a process for enabling connections between non-IDP clients and a printer in an embodiment of the present invention. At step 701, a host requests a connection and an I/O manager increments a pending connection count at step 703. The IDP emulator 40 polls the I/O manager for the current pending connection at step 705 and the IDP emulator generates a CSCP request service port request at step 707. The connection services layer 60 receives a CSCP request at step 709 and a pending open from the IDP server 80 is returned from the connection services 60 via the system services 70 at step 711. The IDP emulator 40 then issues a print job request to the IDP server 80 at step 713 and the IDP server queues a print job at step 715. The IDP emulator 40 then closes the connection to the IDP server 80 at step 717 before the system services 70 dequeues the print job request at step 719 and then issues a connection request at step 721. The connection services layer 60 issues an active open call to the IDP emulator 40 via the socket services layer 50 at step 723 and sends a CSCP request service port to the IDP emulator 40 at step 725. The connection services layer 60 returns from the reconnection request via the

system services layer 70 at step 727 and then the system services layer 70 issues a "send job data" IDP request to the IDP emulator 40 at step 729. The IDP emulator 40 opens a connection to the corresponding I/O manager at step 731 and the corresponding I/O manager accepts the open connection request from the host at step 733. Next, the system services layer 70 sends the I/O stream to the PDL interpreter 90 at step 735. The host sends and receives data to and from the PDL interpreter 90 at step 737 before closing the connection to the host at step 739 when the sending and receiving of data is completed.

Figure 6 illustrates a network configuration of a printing system for an embodiment of the present invention. In Figure 6, a plurality of host computers 400_0, 400_1, ...400_n and a printer 410 are connected in the network. The printer 410 includes a ROM 412, a RAM 414, and a CPU 416. The ROM 412 implements the layers of architecture for interfacing the host computers 400...n with the printer 410 in conjunction with the CPU 416. More than one ROM may be used to implement the layers of the architecture. The RAM 414 is used to implement the print queue 82 in the present embodiment. Because the print queue 82 only stores job information data without storing the actual print data, a small sized RAM 414 is used in contrast to a larger sized spooling disk as is conventionally used.

The print queue 82 contains job information for all of the print jobs that have an outstanding print request to the printer 410 for any of the host computers $400_{0..n}$ on the network. A print job request call is one example of the process for enabling communication with the print queue 82. Prior to sending a print job with a print job request call, the host computer sends a print request and a job ID is returned by the printer. When the print request is accepted, the job information is placed in the print queue 82. The parameters of the print job request call may include information about the print job and information for enabling the printer to notify the host computer when the job data can be accepted. When the printers are ready to capture the print job data, the printer may initiate a transmit job call which tells the host computer to start transmitting the print job to the printer. As print jobs are completed, their job information is removed from the print queue and entered into a job log. The job information in the print queue 82 may include the job name, the estimated time to print, the document status, the number of pages in the document, the number of pages left to print, the image content, the user comment, the media, and the creator application. The print job request call may provide this information and this information must be presented with the job in order to display the job information.

The print queue 82 may also be manipulated by an operator from a local or remote user interface. Examples of queue manipulation commands include interrupt, resume, delete and reorder. In addition, queue manipulation may allow a user to change the job priority and sorting algorithms. For instance, a job sorting algorithm may print all jobs requiring a particular media, such as transparencies first and then, by queue manipulation, the media may be changed and all print jobs needing the newly loaded media may be printed.

The print job request command comes from a host computer to request the transmission of a job to the printer. When a print job request command is accepted, the information for a print job is placed in the print queue and the job ID is returned. The job ID is a unique numerical identifier for the print job and it could be used to reference print jobs and acquire further information and status. The print job request command may involve a dialog between the host computer and the printer to negotiate the connection to send the print job data on. The printer can elect to send back a result code on the response that gives the host computer an opportunity to send the print job data on the same connection that they are currently communicating over. If the host computer chooses not to use this communication channel to send the print job data, a normal transmit job command will be used to solicit the print job data by the printer.

A transmit job command is transmitted from the printer to inform the host computer to start transmitting the print job referred to by the job ID and an enumerate queue command is transmitted from the host computer and returns an ordered list of the jobs in the specified queue. The enumerate queue command counts parameters in the print queue and count records are returned in the order that they are scheduled to be printed for the print queue in the order that they were printed for the job log. A job record parameter may be included for estimating the time when the job will complete printing or the time that the job did complete. A token parameter may be used to represent the state of the queue.

Those skilled in the art will appreciate that IDP provides an application layer network protocol that enhances the two way communication between a host computer and a network printer. IDP operates independently of the network layers below it and only requires that the transport protocol/port be bidirectional. A wide variety of heterogenous network protocols may be supported by IDP which places all of the incoming print job information in the print queue regardless of the protocol. Print job information from both IDP and non-IDP protocol/ports may be placed in the print queue by emulating IDP

on the non-IDP protocol/ports. By emulating IDP on the non-protocol/ports, every client will look like IDP clients to the printer and clients of non-IDP protocols will therefore be completely backward compatible. Although non-IDP clients will not be able to use all of the enhanced features provided by IDP, a sufficient amount of information will provided from the non-IDP protocol/ports so that all of the jobs in the print queue will be able to be fairly arbitrated. For example, complete job name information will not be provided to the print queue for non-IDP clients and only information that a print job has been sent from a particular port will be entered in print queue. Thereby, print jobs from non-IDP clients may be entered in the print queue without having to store the actual print job data at the printer. Accordingly, the printing system for the embodiments of the present invention enable a wide variety of heterogenous network protocols to be fairly arbitrated at the printer.

The network printing system according to the embodiments of the present invention provide seamless plug and play connectivity for various IDP and non-IDP protocol/ports. CSCP facilitates this connectivity by providing a symmetric protocol for connections between servers and clients over connection based-stream oriented protocol such as ADSP and TCP. CSCP resides in the presentation layer of the OSI seven-layer model and interfaces with one or more session level protocols. The dynamic port negotiation of CSCP facilitates concurrency for server applications by off-loading the port negotiation that would otherwise have to take place by the server and its clients. CSCP also allows the printing system to be easily extended and integrated with new protocol/ports as they are developed and introduced.

The invention being thus described, it will be obvious that the same may be varied in many ways. Such variations are not to be regarded as a departure from the spirit and scope of the invention, and all such modifications as would be obvious to one skilled in the art are intended to be included within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A network printing system including a plurality of host computers, said network printing system comprising:

a plurality of network services protocol/port for connecting said plurality of host computers to the network;

a printer connected to the network; and

a connection protocol interface for facilitating connections between said plurality of host computers and said printer over stream oriented protocols.

2. A network printing system according to Claim 1, wherein said connection protocol interface comprises a client server connection protocol (CSCP) for opening an active connection from one of said plurality of host computers to the network, negotiating for a socket to implement said active connection, and establishing the connection between one of said plurality of host computers and said printer.

3. A network printing system according to Claim 1 or 2, wherein said connection protocol interface enables said printer to call back and reconnect with one of said plurality of host computers upon availability of said printer which was previously unavailable to said one host computer.

4. A network printing system according to one of Claims 1-3, wherein said stream oriented protocols comprise AppleTalk Data Stream Protocol (ADSP), transmission connection protocol (TCP) and Novell sequence packet exchange protocol (SPX).

5. A method for providing connections in a network printing system comprising the steps of:

(a) connecting a plurality of host computers to a printer in a network by a plurality of network services protocol/ports; and

(b) facilitating connections between said host computers and said printer over stream oriented protocols.

6. A method according to Claim 5, wherein said step (b) further comprises the step of providing a client server connection protocol (CSCP) for facilitating connections including the steps of opening an active connection from one of said host computers to the network, negotiating for a socket to implement said

active connection, and establishing the connection between said one host computer and said printer.

7. A method according to Claim 5 or 6, wherein said step (b) further comprises the step of enabling said printer to call back and reconnect with one of said host computers, upon availability of said printer which was previously unavailable to said one host computer.

8. A method according to one of Claims 5-7, wherein said stream oriented protocols comprise AppleTalk Data Stream Protocol (ADSP), transmission connection protocol (TCP) and Novell sequence packet exchange protocol (SPX).

9. An architecture for a network printing system comprising:

a plurality of network service protocol/ports for interfacing a plurality of host computers to a network printer, said network service protocol/ports including imaging device protocol (IDP) ports and non-IDP ports;

an IDP emulator for emulating IDP connection requests when a connection request is received from one of said non-IDP ports;

a socket services layer for interfacing with said network service protocol/ports to request and receive information;

a connection services layer for facilitating connections between said network service protocol/ports and the network printer over connection based-stream oriented protocols;

a system services layer communicative with said connection services layer for providing a uniform interface for said IDP and non-IDP ports;

an IDP server communicative with said system services layer for enabling two-way communication between said printer, host computers and the network; and

a page description language (PDL) interpreter communicative with said IDP server and said system services layer for submitting print data from said host computers to the network printer.

10. An architecture according to Claim 9, wherein said non-IDP ports comprise printer access protocol (PAP)/LocalTalk ports, PAP/EtherTalk ports, transmission control protocol (TCP)/internet protocol (IP) ports, LPR/TCP/IP

ports, Novell remote printer/printer server mode ports, and bi-directional or unidirectional Centronics ports.

11. An architecture according to Claim 9 or 10, wherein said IDP ports comprise AppleTalk Data Stream Protocol (ADSP) ports, ADSP/LocalTalk ports, Novell sequence pocket exchange (SPX) ports, TCP/IP ports, or bi-directional Centronics ports.

12. An architecture according to one of Claims 9-11, wherein said IDP emulator uniformly interfaces said IDP and non-IDP ports to said connection services layer, said system services layer, said IDP server and said PDL interpreter.

13. An architecture according to one of Claims 9-12, wherein said IDP emulator generates an IDP print job request in response to a connection request from one of said non-IDP ports.

14. A printer for a network printing system, said printer comprising:

a plurality of network service protocol/ports for interfacing a plurality of host computers to the printer, wherein said network service protocol/ports are comprised of imaging device protocol (IDP) ports and non-IDP ports;

an IDP emulator for emulating IDP connection requests when a connection request is received from one of said non-IDP ports;

a socket services layer for interfacing with said network service protocol/ports to request and receive information;

a connection services layer for facilitating connections between said network service protocol/ports and the printer over connection based stream oriented protocol;

a system services layer communicative with said connection services layer for providing a uniform interface for said IDP and non-IDP ports;

an IDP server communicative with said system services layer for enabling two-way communication between said printer, host computers and the network; and

a page description language (PDL) interpreter communicative with said IDP server and said system services layer for submitting print data from said host computers to the printer.

15. A method for interfacing with a network printing system, comprising the steps of:

(a) interfacing network host computers to the network printing system with network service protocol/ports including imaging device protocol (IDP) ports and non-IDP ports;

(b) emulating IDP connection requests when a connection request is received from one of said non-IDP ports;

(c) interfacing said network service protocol/ports with a socket services layer to receive and request information;

(d) facilitating connections between said network service protocol/ports and the network printing system over connection based-stream oriented protocols with a connection services layer;

(e) providing a uniform interface for said IDP and said non-IDP ports by a system services layer communicative with said connection services layer;

(f) enabling two-way communication between said network host computers and the network printing system with an IDP server communicative with said system services layer; and

(g) submitting print data from said network host computers to the network printer of the printing system by a page description language (PDL) interpreter communicative with said IDP server and said system services layer.

16. A method according to Claim 15, wherein said step (b) uniformly interfaces said IDP and non-IDP ports to said connection services layer, said system services layer, said IDP server and said PDL interpreter.

17. A method according to Claim 15 or 16, wherein said step (b) comprises the step of generating a IDP print job request in response to a connection request from one of said non-IDP ports.

18. A computer readable medium comprising program instruction layers for:

(a) interfacing network host computers to the network printing system with network service protocol/ports including imaging device protocol (IDP) ports and non-IDP ports;

(b) emulating IDP connection requests when a connection request is received from one of said non-IDP ports;

(c) interfacing said network service protocol/ports with a socket services layer to receive and request information;

(d) facilitating connections between said network service protocol/ports and the network printing system over connection based-stream oriented protocols with a connection services layer;

(e) providing a uniform interface for said IDP and said non-IDP ports by a system services layer communicative with said connection services layer;

(f) enabling two-way communication between said network host computers and the network printing system with an IDP server communicative with said system services layer; and

(g) submitting print data from said network host computers to the network printer of the printing system by a page description language (PDL) interpreter communicative with said IDP server and said system services layer.

19. A network printing system including at least one host computer, said network printing system comprising:

at least one network services protocol/ports for connecting said at least one host computer to the network;

a printer connected to the network; and

an interface for uniformly interfacing from said network services protocol/ports to said printer.

20. A network printing system according to Claim 19, wherein said interface comprises an emulator for emulating a uniform interfacing protocol to said printer from each of said network services protocol/ports.

21. A network printing system according to Claim 19, wherein said interface comprises an IDP emulator for emulating IDP requests to said printer from said non-IDP ports so that both IDP and non-IDP requests may be queued to said printer.

22. A network printing system according to one of Claims 19 to 21, wherein said network services protocol/ports comprise imaging device protocol (IDP) ports and non-IDP ports.

23. A method for providing a uniform interface in a network printing system comprising the steps of:

(a) connecting a plurality of host computers to a printer in a network by a plurality of network services protocol/ports; and

(b) communicating via a uniform interfacing protocol for said network services protocol/ports to said printer.

24. A method according to Claim 23, wherein said step (b) comprises the step of emulating said uniform interfacing protocol to said printer from each of said network services protocol/ports.

25. A method according to Claim 23, wherein said step (b) comprises the step of emulating IDP requests to said printer from said non-IDP ports so that both IDP and non-IDP requests may be queued to said printer.

26. A method according to one of Claims 23 to 25, wherein said network services protocol/ports comprise IDP ports and non-IDP ports.

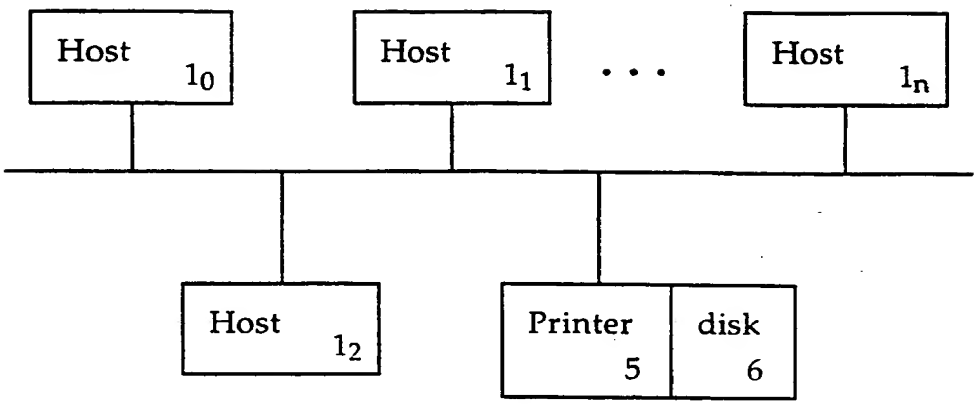

Host 1₀
Host 1₁
Host 1ₙ
Host 1₂
Printer 5
disk 6

Fig. 1

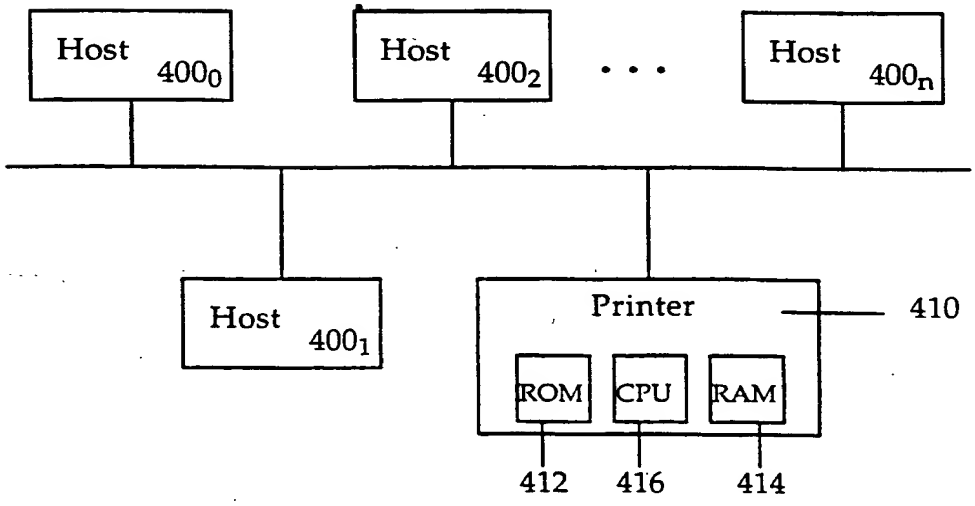

Host 400₀
Host 400₂
Host 400ₙ
Host 400₁
Printer
410
ROM
CPU
RAM
412
416
414

Fig. 6

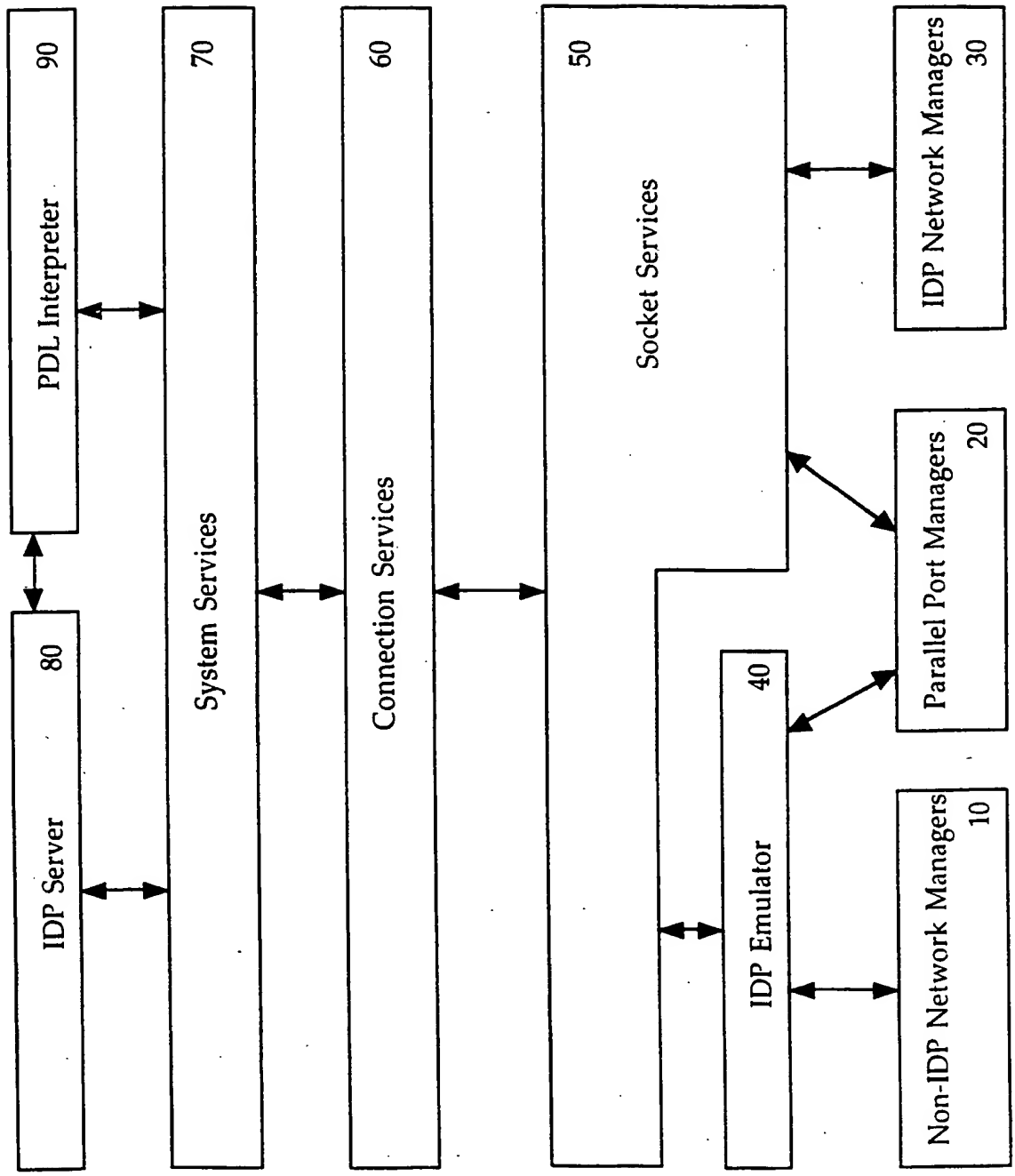

PDL Interpreter 90
IDP Server 80
System Services 70
Connection Services 60
Socket Services 50
IDP Emulator 40
IDP Network Managers 30
Parallel Port Managers 20
Non-IDP Network Managers 10

Fig. 2

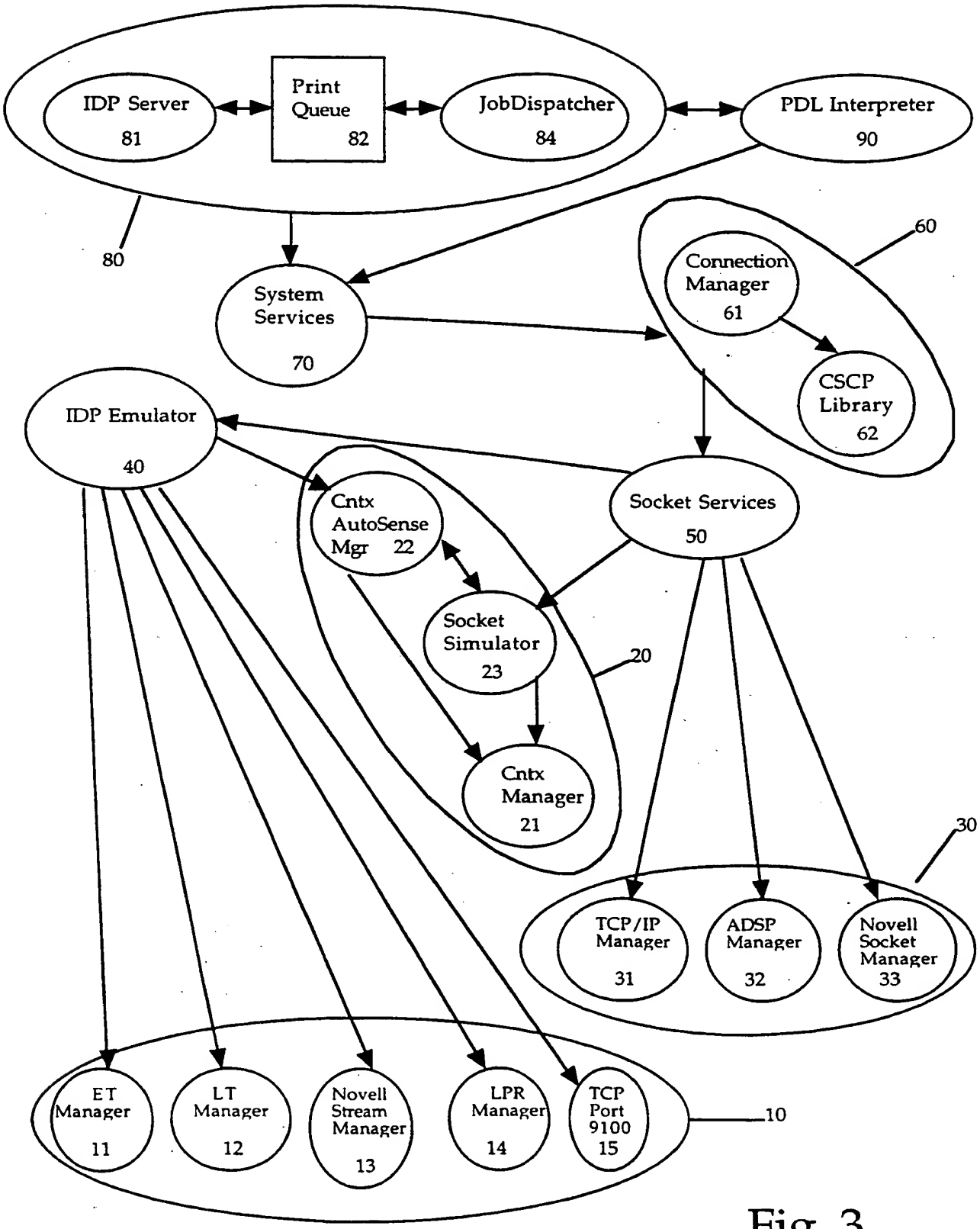


Fig. 3

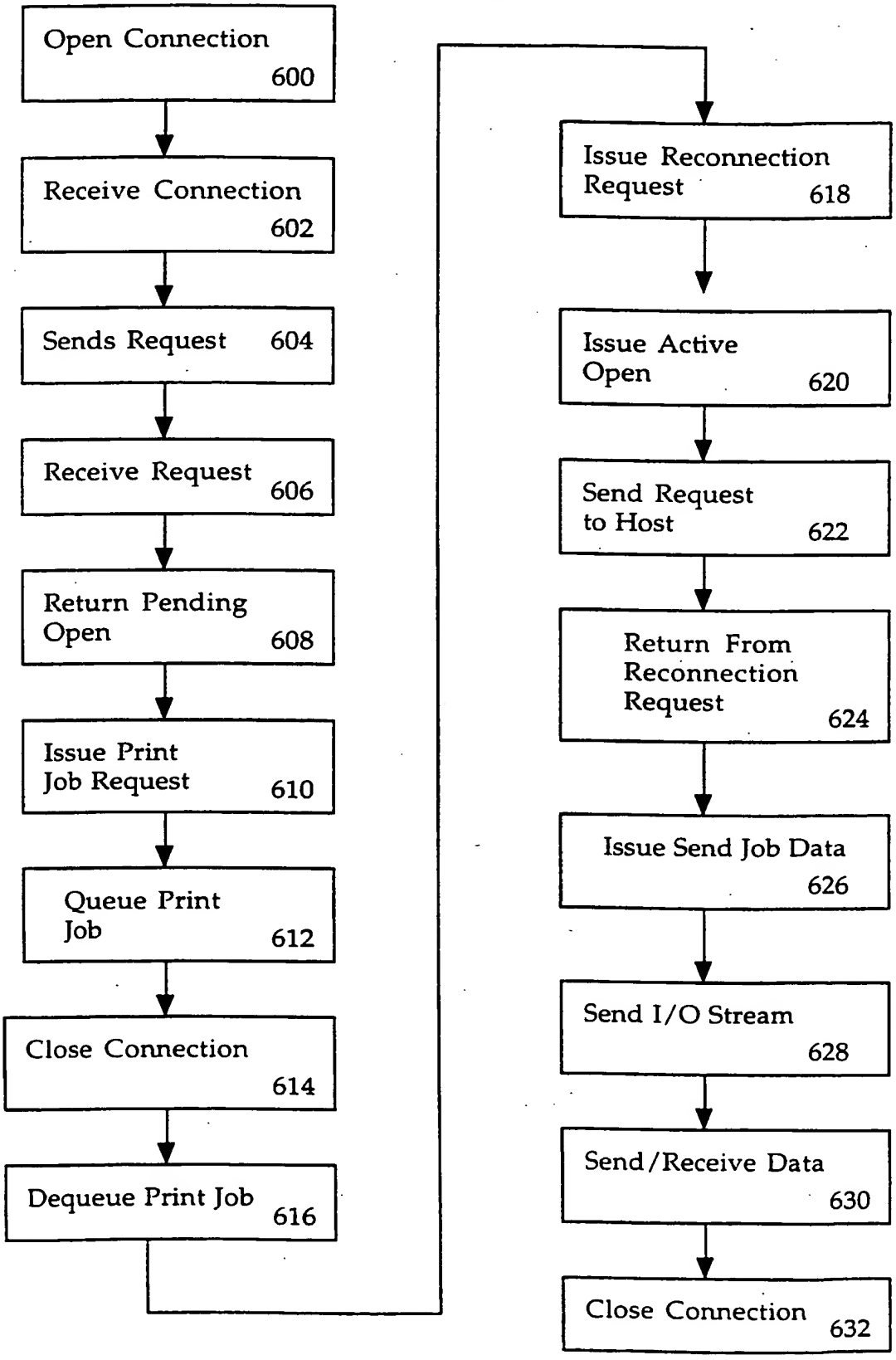


Fig. 4

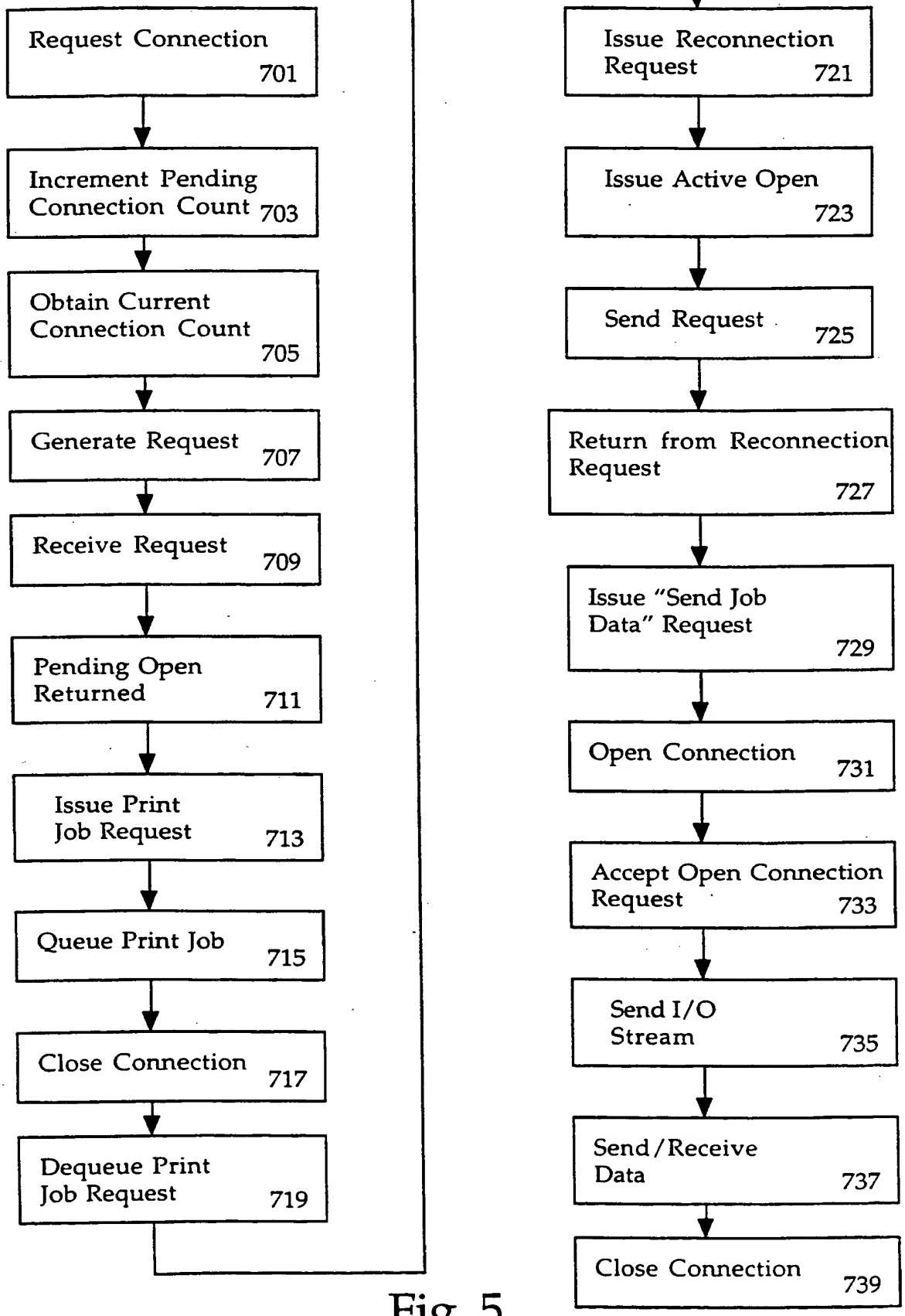


Fig. 5

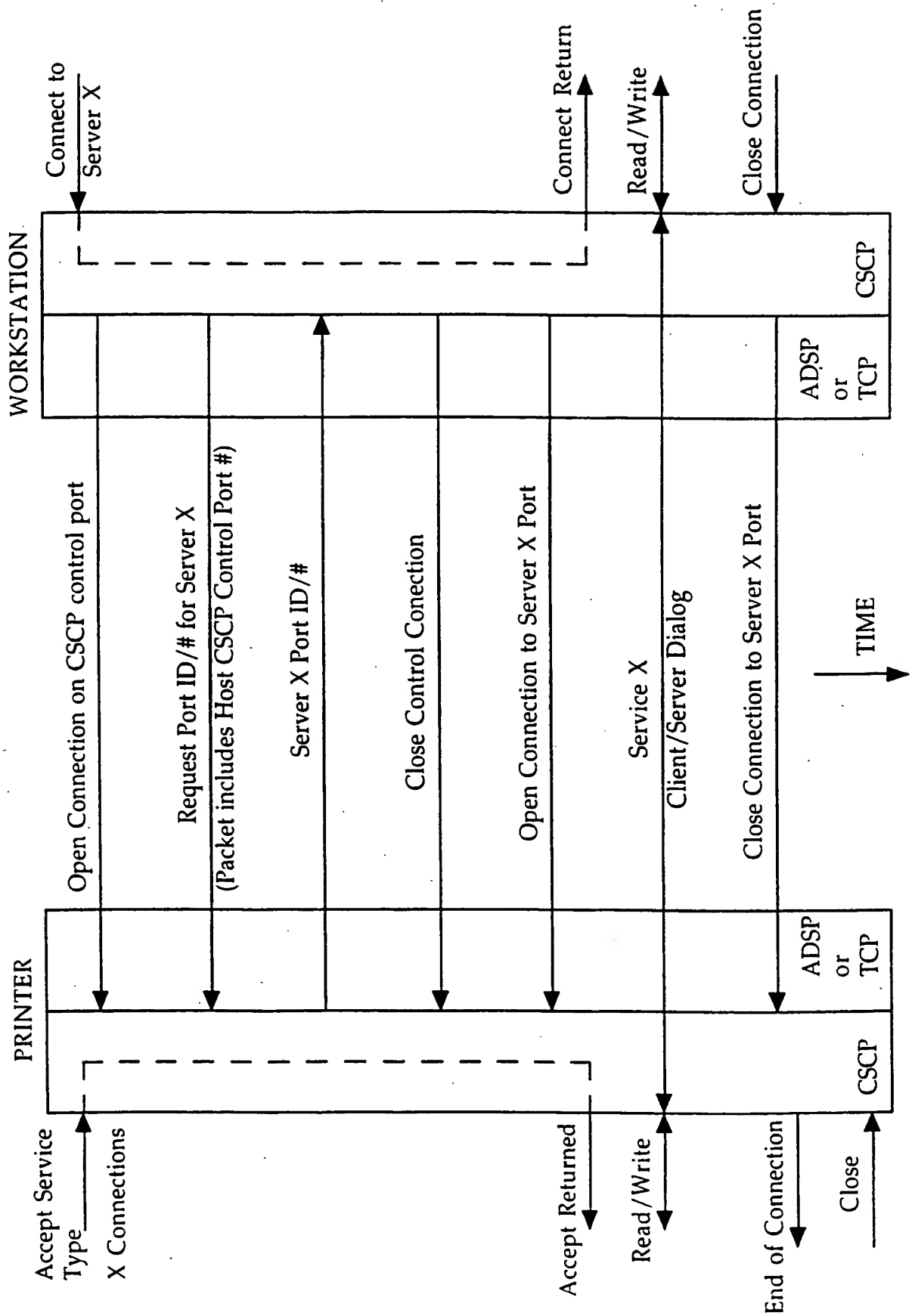


Fig. 7

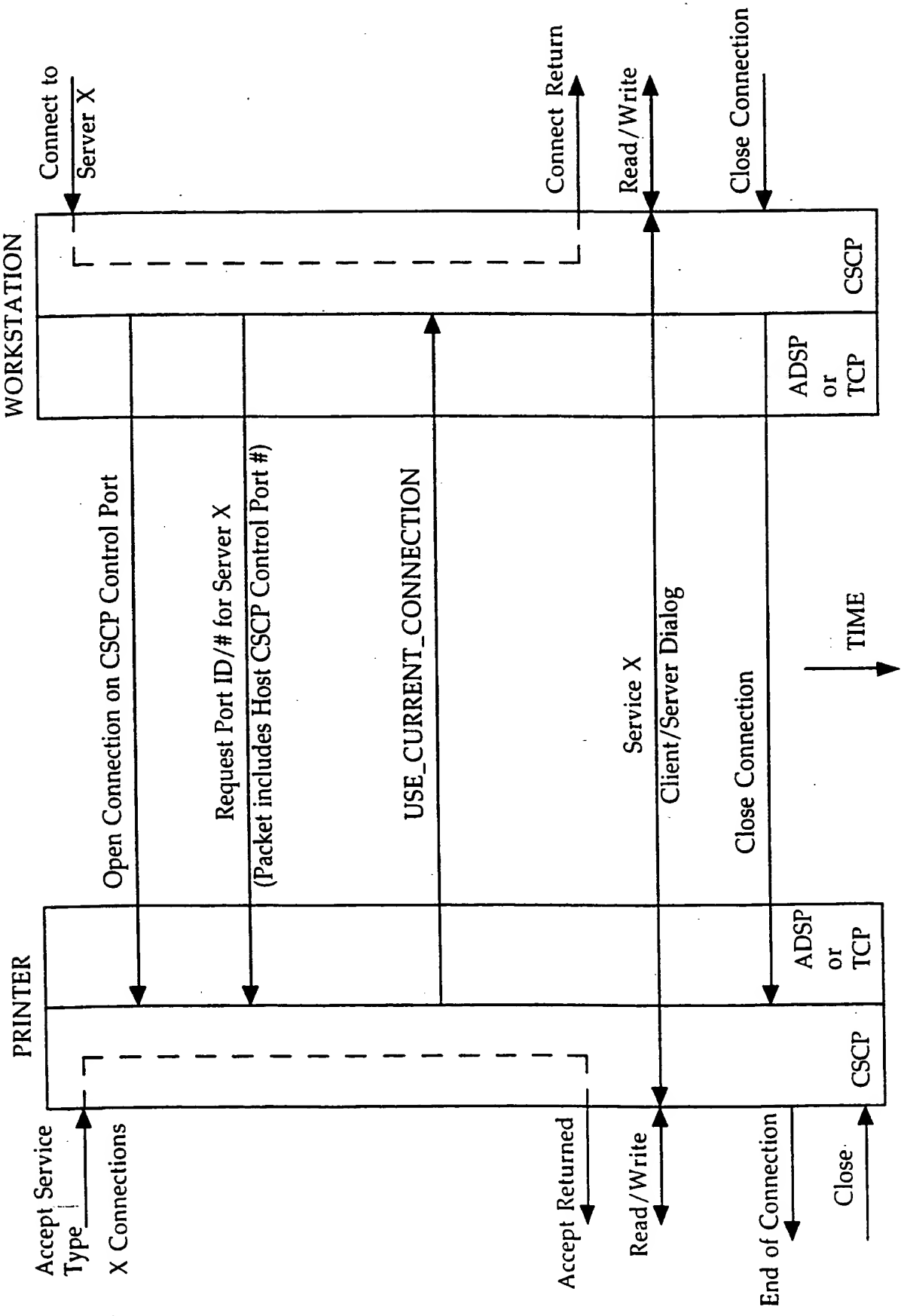


Fig. 8

INTERNATIONAL SEARCH REPORT

In. ational Application No
PCT/US 96/08491

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F3/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP,A,0 598 511 (CANON INFORMATION SYST INC) 25 May 1994 see figures 1-3,5,11-13 see page 4, line 3 - line 56 see page 6, line 27 - page 7, line 9 see page 12, line 42 - page 13, line 48	1-7, 9-16,18, 19,22, 23,25,26
A	EP,A,0 653 700 (FUJITSU LTD) 17 May 1995 see figures 1,2,5C,10,15 see column 7, line 55 - column 11, line 26 see column 12, line 3 - line 25	1,5,6, 8-15, 18-20, 23-25

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

2

Date of the actual completion of the international search	Date of mailing of the international search report
29 October 1996	06. 11. 96
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax (+ 31-70) 340-3016	Authorized officer Weiss, P

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 96/08491

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0598511	25-05-94	JP-A- 6214929	05-08-94
EP-A-0653700	17-05-95	JP-A- 7141132	02-06-95
		JP-A- 7261954	13-10-95